UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/777,909	02/12/2004	William Preston Alexander III	AUS920030825US1	6082

35525 7590 01/13/2012

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
RAMPURIA, SATISH	

ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
01/13/2012	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptonotifs@yeeiplaw.com

Office Action Summary	Application No.	Applicant(s)	
	10/777,909	ALEXANDER ET AL.	
	Examiner	Art Unit	
	SATISH RAMPURIA	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 December 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *1 and 4-8* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *1 and 4-8* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12/08/2011*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the RCE filed on 12/08/2011.

2. Claims cancelled by the applicants: 2-3 and 9-24.

3. Claims 1 and 4-8 are pending.

Continued Examination Under 37 CFR 1.114

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 12/08/2011 has been entered.

Information Disclosure Statement

5. An initialed and dated copy of Applicant's IDS form 1449 filed on 12/08/2011 is attached to the instant Office action.

For items 12 and 13, no office action was found under the specified docket number nor the specified date of the office action sent out. Thus, items 12 and 13 were not considered.

Oath/Declaration

6. The Office acknowledges receipt of a properly signed oath/declaration filed 02/12/2004.

Drawings

7. The drawings were received on 02/12/2004. These drawings are acceptable by the examiner.

Double Patenting

8. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

9. Claim 1 and 4-7 provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 and 4-7 of copending Application No. 12/172,791. Although the conflicting claims are not identical, they are not patentably distinct from each other because of following observation.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Instant Claim	12/172,791 Claim
1. (Original) A method, in a data processing system, for generating a minimized call tree data structure from trace data obtained from a plurality of executions of a computer program,	1. A method, in a data processing system, for automatically identifying performance regression between builds of a computer program based on trace data obtained from a

comprising: obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program; generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; and outputting the minimized call tree data structure.	plurality of executions of a first and second build of a computer program, comprising: obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the first and second builds of the computer program; generating a minimized call tree data structure from the plurality of call tree data structures for each of the first and second builds of the computer program, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; subtracting the minimized call tree data structure for the second build of the computer program from the minimized call tree data structure of the second computer program to thereby generate a subtracted minimized call tree data structure; and outputting the subtracted minimized call tree data structure.
4. (Original) The method of claim 1, wherein generating the minimized call tree data structure includes: copying a first call tree data structure; and walking a second call tree data structure over the first call tree data structure to generate the minimized call tree data structure.	4. The method of claim 1, wherein generating the minimized call tree data structure includes: copying a first call tree data structure for a selected build of the computer program; and walking a second call tree data structure for the selected build of the computer program over the first call tree data structure to generate the minimized call tree data structure.
5. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node.	5. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node.
6. (Original) The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values	6. The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values

associated with corresponding nodes in the first call tree data structure and the second call tree data structure.	associated with corresponding nodes in the first call tree data structure and the second call tree data structure.
7. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in only one of the first call tree data structure and the second call tree data structure, inhibiting creating a node in the minimum call tree data structures.	7. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in only one of the first call tree data structure and the second call tree data structure, inhibiting creating a node in the minimum call tree data structures.

10. Claims 1 and 4-7 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1 of USPN 7,496,900. Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observation.

Instant Claim	7,496,900 Claim
1. (Original) A method, in a data processing system, for generating a minimized call tree data structure from trace data obtained from a plurality of executions of a computer program, comprising: obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program; generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; and outputting the minimized call tree data structure.	1. A method, in a data processing system, for automatically identifying performance regression between builds of a computer program based on trace data obtained from a plurality of executions of a first and second build of the computer program, comprising: **obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the first and second builds of the computer program;** **generating a minimized call tree data structure from the plurality of call tree data structures for each of the first and second builds of the computer program, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree**

4. (Original) The method of claim 1, wherein generating the minimized call tree data structure includes: copying a first call tree data structure; and walking a second call tree data structure over the first call tree data structure to generate the minimized call tree data structure. 5. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node. 6. (Original) The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structure. 7. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in only one of the first call tree data structure and the second call tree data structure, inhibiting creating a node in the minimum call tree data structures.	**data structures, wherein generating the minimized call tree data structure further comprises: copying a first call tree data structure for a selected build of the computer program; and walking a second call tree data structure for the selected build of the computer program over the first call tree data structure to generate the minimized call tree data structure, wherein walking the second call tree data structure over the first call tree data structure further comprises: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node, wherein the values associated with the node are values that correspond to a minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structures; and for each node that exists in only one of the first call tree data structure and the second call tree data structure, inhibiting creating a node in the minimized call tree data structure;** subtracting the minimized call tree data structure for the second build of the computer program from the minimized call tree data structure of the second computer program to thereby generate a subtracted minimized call tree data structure, wherein subtracting the minimized call tree data structure for the second build from the minimized call tree data structure for the first build to generate a subtracted minimized call tree data structure further comprises: copying the minimized call tree data structure for the first build; and walking the minimized call tree data structure for the second build over the minimized call tree data structure for the first build to generate the subtracted minimized call tree data structure, wherein walking the minimized call tree data structure for the second build over the

	minimized call tree data structure for the first build further comprises: for each node that exists in both the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, generating a node in the subtracted minimized call tree data structure by subtracting a minimum base value of the node in the minimized call tree data structure for the second build from a minimum base value of a corresponding node in the minimized call tree data structure for the first build; and for each node that exists in only one of the minimized call tree data structure for the first build and the minimized call tree data structure for the second build, creating a node in the subtracted minimized call tree data structure having a negative minimum base value corresponding to a minimum base value of the node that exists in either of the minimized call tree data structure for the first build or the minimized call tree data structure for the second build; **outputting the subtracted minimized call tree data structure**; and inputting the trace data to an arcflow tool, wherein the arcflow tool generates the plurality of call tree data structures for each of the first and second builds of the computer program based on the trace data.

11. Claims 1, 4-6 and 8 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 4-6, 8, 9, 12, 13, 15-17 and 20-22 of USPN 7,519,961. Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observation.

Instant Claim	7,519,961Claim
1. (Original) A method, in a data processing system, for generating a minimized call tree data structure from trace data obtained from a plurality of executions of a computer program, comprising: obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program; generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; and outputting the minimized call tree data structure.	1. A method, in a data processing system, for averaging out variations in trace data obtained from a plurality of executions of a computer program, comprising: obtaining call tree data structures corresponding to the trace data for the plurality of executions of the computer program; adding the call tree data structures to generate an added call tree data structure; calculating an average of values associated with each node in the added call tree data structure to generate an averaged call tree data structure; and outputting the averaged call tree data structure, wherein the affect of variations in trace data of various executions of the computer program are minimized in the averaged call tree data structure.
4. (Original) The method of claim 1, wherein generating the minimized call tree data structure includes: copying a first call tree data structure; and walking a second call tree data structure over the first call tree data structure to generate the minimized call tree data structure.	4. The method of claim 1, wherein adding the call tree data structures to generate an added call tree data structure includes: copying a first call tree data structure; and walking a second call tree data structure over the first call tree data structure to generate the added call tree data structure.
5. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values	5. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the added call tree data

with the node. 6. (Original) The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structure. 8. (Original) The method of claim 6, wherein the values associated with each node in the minimized call tree data structure include a minimum base value, a minimum number of calls, a minimum cumulative value, and a minimum absolute cumulative value.	structure by adding a base value of the node in the second call tree data structure to a base value of a corresponding node in the first call tree data structure. 6. The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in only one of the first call tree data structure and the second call tree data structure, creating a node in the added call tree data structure having a base value corresponding to the base value of the node that exists in either of the first call tree data structure or the second call tree data structure. 8. The method of claim 1, wherein the values associated with each node include a base value, a number of calls, a cumulative value, and an absolute cumulative value.
1. (Original) A method, in a data processing system, for generating a minimized call tree data structure from trace data obtained from a plurality of executions of a computer program, comprising: obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program; generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; and outputting the minimized call tree data structure.	9. A computer program product in a computer readable medium for averaging out variations in trace data obtained from a plurality of executions of a computer program, comprising: first instructions for obtaining call tree data structures corresponding to the trace data for the plurality of executions of the computer program; second instructions for adding the call tree data structures to generate an added call tree data structure; third instructions for calculating an average of values associated with each node in the added call tree data structure to generate an averaged call tree data structure; and fourth instructions for outputting the averaged call tree data structure, wherein the affect of variations in trace data of various executions of the computer program are minimized in the averaged call tree data structure.

4. (Original) The method of claim 1, wherein generating the minimized call tree data structure includes: copying a first call tree data structure; and walking a second call tree data structure over the first call tree data structure to generate the minimized call tree data structure.	12. The computer program product of claim 9, wherein the second instructions for adding the call tree data structures to generate an added call tree data structure include: instructions for copying a first call tree data structure; and instructions for walking a second call tree data structure over the first call tree data structure to generate the added call tree data structure.
5. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node.	13. The computer program product of claim 12, wherein the instructions for walking the second call tree data structure over the first call tree data structure include: for each node that exists in both the first call tree data structure and the second call tree data structure, instructions for generating a node in the added call tree data structure by adding a base value of the node in the second call tree data structure to a base value of a corresponding node in the first call tree data structure.
6. (Original) The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structure.	15. The computer program product of claim 9, wherein the third instructions for calculating an average of values associated with each node in the added call tree data structure to generate an averaged call tree data structure include: instructions for dividing values associated with each node in the added call tree data structure by a total number of call tree data structures that were added to generate the added call tree data structure.
8. (Original) The method of claim 6, wherein the values associated with each node in the minimized call tree data structure include a minimum base value, a minimum number of calls, a minimum cumulative value, and a minimum absolute cumulative value.	16. The computer program product of claim 9, wherein the values associated with each node include a base value, a number of calls, a cumulative value, and an absolute cumulative value.
1. (Original) A method, in a data processing system, for generating a minimized call tree data structure from trace data obtained from a plurality of executions of a computer program,	17. An apparatus for averaging out variations in trace data obtained from a plurality of executions of a computer program, comprising: means for obtaining call tree data structures

comprising: obtaining a plurality of call tree data structures corresponding to the trace data for the plurality of executions of the computer program; generating a minimized call tree data structure from the plurality of call tree data structures, wherein the minimized call tree data structure includes a minimum set of nodes that are consistent between the plurality of call tree data structures; and outputting the minimized call tree data structure.	corresponding to the trace data for the plurality of executions of the computer program; means for adding the call tree data structures to generate an added call tree data structure; means for calculating an average of values associated with each node in the added call tree data structure to generate an averaged call tree data structure; and means for outputting the averaged call tree data structure, wherein the affect of variations in trace data of various executions of the computer program are minimized in the averaged call tree data structure.
4. (Original) The method of claim 1, wherein generating the minimized call tree data structure includes: copying a first call tree data structure; and walking a second call tree data structure over the first call tree data structure to generate the minimized call tree data structure.	20. The apparatus of claim 17, wherein the means for adding the call tree data structures to generate an added call tree data structure includes: means for copying a first call tree data structure; and means for walking a second call tree data structure over the first call tree data structure to generate the added call tree data structure.
5. (Original) The method of claim 4, wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the minimized call tree data structure and associating values with the node.	21. The apparatus of claim 20, wherein the means for walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, means for generating a node in the added call tree data structure by adding a base value of the node in the second call tree data structure to a base value of a corresponding node in the first call tree data structure.
6. (Original) The method of claim 5, wherein the values associated with the node are values that correspond to the minimum of the values associated with corresponding nodes in the first call tree data structure and the second call tree data structure.	22. The apparatus of claim 20, wherein the means for walking the second call tree data structure over the first call tree data structure includes: for each node that exists in only one of the first call tree data structure and the second call tree data structure, means for creating a node in the added call tree data structure having a base value corresponding to

	the base value of the node that exists in either of the first call tree data structure or the second call tree data structure.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Satish Rampuria whose telephone number is (571) 272-3732. The examiner can normally be reached on 8:30 am to 5:00 pm Monday to Friday. Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Satish Rampuria/
Examiner, Art Unit 2191

/Anna Deng/

Primary Examiner, Art Unit 2191